مقدمة فى لغة

F# SHARP

ال F# اصبحت لغة رسميه من لغات ال dotnet ومن ضمن لغات ال Visual Studio 2010

(كتاب مجمع من مصادر و مقالات متعددة)

الإصدار الاول

اعداد م \ أيمن محمد عبد العزيز عبد الله تميم

Ayman78007@hotmail.com

الاثنين ١٢ ربيع اول لعام ١٤٣٠ هــ الموافق ٩-٣-٢٠٠٩

فهرس الكتاب

مثال لكود مكتوب بلغة F#

```
open System
open System.Windows
open System.Windows.Controls
open System.Windows.Markup
open System.Windows.Media
open System.Xml
(*************************************************************)
let rec dumpVisualTree level (elem : DependencyObject) =
  // Write out this element's info.
  let indent = new String(' ', level)
  let name = elem.GetType().Name
  Console.WriteLine("{0}{1}) {2}", indent, level, name)
  // Walk down to this element's children.
  let numChildren = VisualTreeHelper.GetChildrenCount(elem)
  for index = 0 to numChildren - 1 do
    let child = VisualTreeHelper.GetChild(elem, index)
    dumpVisualTree (level + 1) child
(*************************************************************)
// Curry the function, just for kicks.
let startDump = dumpVisualTree 1
(*************************************************************)
let createWindow (file : string) =
  // Notice the 'use' keyword here, which is like a C# using()
  // block.  Thanks to Robert Pickering for pointing that out.
  use xmlRdr = XmlReader.Create(file)
  let wnd = XamlReader.Load(xmlRdr) :?> Window
  wnd.Loaded.Add (fun _ -> startDump(wnd :> DependencyObject))
  wnd
(*************************************************************)
let main() =
  let app = new Application()
  app.Run(createWindow("MainWindow.xaml")) |> ignore
  Console.ReadKey() |> ignore
(*************************************************************)
[<STAThread>]
do main()
```

مقدمة منقولة من مبرمجين عرب

مقدمة (منقولة من منتدى فيجويل بيسك بقلم المشرف Mohamed Gamal El-Din)

(مقدمة رقم ١)

بسم الله الرحمن الرحيم

اصدرت شركه ميكروسوفت في الاونه الاخيره بعض الاصدارات الاولي للغة البرمجه الجديده F#

فدعونا الان نفسر ما هي التقنيات الحديثه في تلك اللغه والتي تجعلها من احسن بل احسن لغات البرمجه الموجوده الان

بدأ don syme وهو المسئول عن تصميم لغه ال F# التفكير في بناء لغه جديده تشمل ما يميز كل لغه برمجه علي حده، لكي تكون هناك لغه واحده تستطيع ان تغني عن العشرات من لغات البرمجه المختلفه.

مميزات لغه الF#

يمكن لل F# أن تكون لغه ثابته "Static" او متغيره Dynamic في نفس الوقت وقد اخذت صفه ال Dynamic من لغه ال Python.

التصويرات التفاعليه لبيئة البيانات مثل الMATLAB

لغه برمجه تعتمد علي الوظائف(Functional Programming language) مثل لغة الML

استخدام بيئه الدوت نت (dotnet framework) مثل السي شارب او الفيجوال بيسيك دوت نت

توضيح ملامح الكود (profiling)مثل لغه السي شارب

يقوم (start up)البرنامج في وقت قصير جدا عن طريق استخدام الاداه. NGEN.exe

يمكن لل F# التداخل (integration)من لغات الدوت نت الاخري

هذه كانت نبذه بسيطه عن لغه ال F# وسنكتب عنا بالتفصيل في الايام القادمه بأذن الله.

(ردود على المقدمة رقم ١ من المشرف !!! BoriS !!!)

جزاك الله خيرا اخى

لى تعليق اذا سمحت...

ال Ngen هو عبارة عن مترجم اخر مثل Jit ولكنة يقوم بترجمة البرنامج من أولة الى أخرة دفعة واحدة وهو أختصار الى Native Image Generator ويسمى Pre-JIT Compiler

(مقدمة رقم ٢مثال منقول من المشرف Boutemine Oualid)

السلام عليكم و رحمة الله تعالى وبركاته
لغة البرمجة هذه بسيطة التعلم و هي جد ممتعة وقد استخلصت من لغة ال Objective Camel و هي خاصة بصفة أساسية للرياضيين و الفيزيائيين و لا أنصح بها الإخوة (أنا تعلمت ال O Caml و لذلك أقدم لكم النصيحة)
سأضع بعض الدروس عن هذه اللغة في الأيام القادمة ان شاء الله تعالى
و هنا الدرس الأول: بعض الأساسيات
-1لغة ال F# حساسة لحالة الأحرف.
-2لغة وضائفية تعتمد بشكل أساسي على البرمجة الوضائفية و ليس الكائنية التوجه.
-3جميع الأسطر البرمجية تنتهي بالرمز;;
-4لغة برمجة ديناميكية.
-5بعض القواعد الأساسية:
-لتعريف متغير نبدأ دائما ب let ثم اسم المتغير و بعدها يمكن لنا أن نظيف القيمة الابتدائية أولا
-ال select case في الفيبي نت (switch في السي شارب) تكتب على الشكل التالي
رمز:

```
mutch [variable] with
case1 -> action1
| case2 -> action2
| n-> actionx;;
```

حيث mutch توافقselect case
[variable]المتغير الذي نعمل عليه المقارنة.
case1:الحالة الأولى (القيمة التي نقارن بها)
->:مثل ال : في السي شارب
actionxالعملية التي نقوم بها في حالة استيفاء الشرط
|مثل or أو elseif كما تشاء.
nفي الحالات الأخرى مثل case else في الفيبي نت أو default في السي شارب.

الحلقات التكرارية
رمز:

```
for x=1 to 10 do
....
done;;
```

رمز:

```
while condition do
....
done;;
```

التفرعات
رمز:

```
if condition then
begin
...;;
...;;
end
else
begin

end;;
```

تعريف الدوال
رمز:

```
let functionName x y z =
begin
...
end;;
```

حيث x y z هي براميترات عادية

الدوال التراجعية (الترددادية)

```
let rec FunctionName x y z = begin
end;;
```

مقدمة منقولة من موقع مايكروسوفت و مواقع اجنبية اخرى

تعريف للغة (اف شارب F#) من مواقع متعددة(بدون ترجمة)

الموقع \ المصدر	التعريف
msdn.microsoft.com	**F# is a functional programming language for the .NET Framework. It combines the succinct, expressive, and compositional style of functional programming with the runtime, libraries, interoperability, and object model of .NET**
FORUM	The Origin of F# At the heart of F# lies Objective Caml, or OCaml, and the .NET language C#. C#, as you probably already know, is based upon C/C++ and Java. OCaml is the latest incarnation of the Caml language. Caml is a dialect of ML. F# is, in essence, another dialect, or "manifestation" as Microsoft calls it, of a functional ML-like language mixed with the power of C#. What is ML? ML is a functional language (similar to Scheme, as we discussed above) created about 20 years ago, primarily for research. ML is short for "meta-language", and as the name implies ML is primarily used to manipulate other languages. So a compiler, for example, is a great application to write using ML. ML creates an entirely different development environment, including interactive compilers that respond as you code. A very different mode of thinking is required to code in a functional language. Now, it's not LSD for developers, it's fairly mind-blowing.
F# in 20 Minutes - Part I	**F# is a functional programming language built on .NET. Just like C# and VB.NET, F# can take advantage of core libraries such as Windows Presentation Foundation, Windows Communication Foundation, Visual Studio Tools for Office, etc. With F# you can even write XBox games using XNA.** **But just because you can write code in a new language doesn't mean you should. So why use F#? Because being a functional language, F# makes writing some classes of programs much easier than its imperative cousins like C#. Parallel Programming and Language-Oriented Programming are two such domains that can be expressed easily in F#.**

متطلبات التحميل و التشغيل

Brief Description
F# is a type-safe, scalable language for the .NET platform that supports both functional and object-oriented programming. This CTP release includes the F# compiler and tools, as well as Visual Studio 2008 integration for F# development. Microsoft F#, Community Technology Preview

Quick Details
Version: 1.9.6.2
Date Published: 9/5/2008
Language: English
Download Size: 13.4 MB - 13.9 MB*
***Download size depends on selected download components.**

System Requirements

Supported Operating Systems: Windows Vista; Windows XP

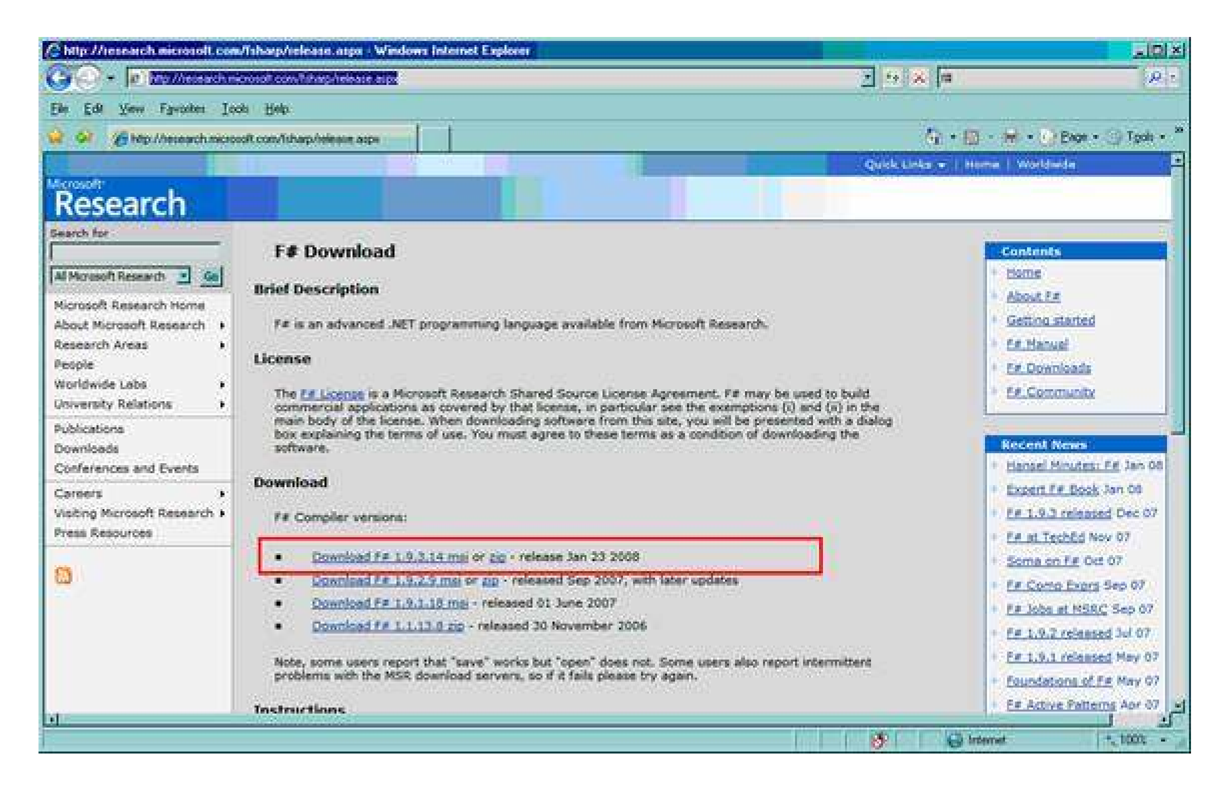
Required software: .NET Framework 2.0

Optional software: Visual Studio 2008 or the Visual Studio 2008 Shell.

تنصيب لغة (F#) على (VISUAL STUDIO 2008)

١- قم بتحميل اخر اصدار من f# compiler من موقع مايكروسوفت على الوصلة التالية

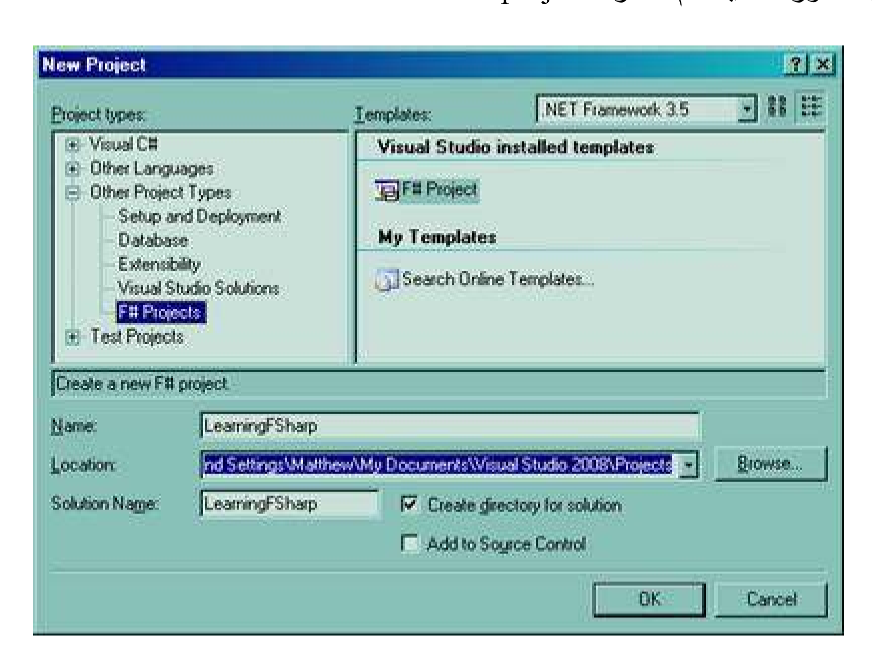
http://research.microsoft.com/en-us/um/cambridge/projects/fsharp/release.aspx



٢-للبدء ادخل على (visual studio 2008) بعدما قمت بتحميل ال(f# compiler) على الوندوز الخطوة الاولى .

٣- اختار مشروع جديد اختار

("Other Project Types") كما فى الصورة التالية ثم اختار f# projects

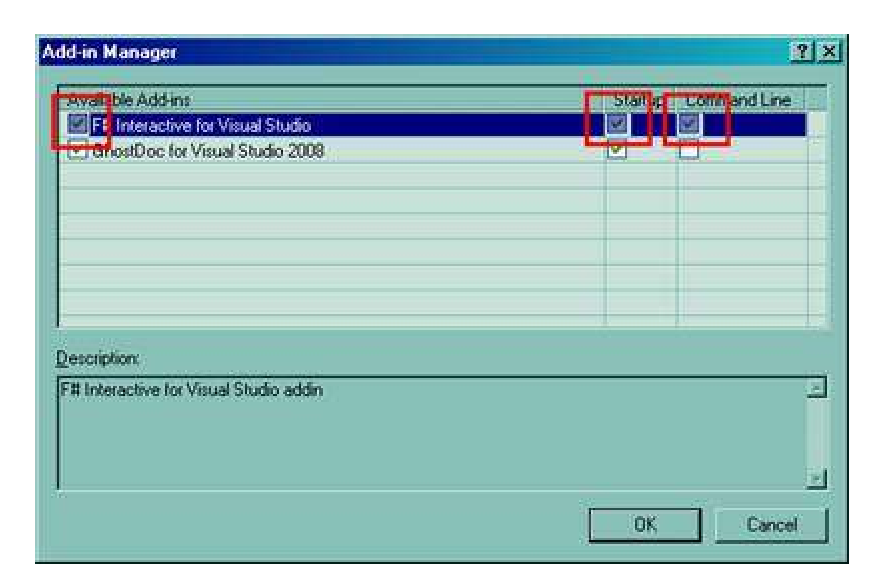


٤- بنجاحك فى بدئ مشروع جديد بلغة f# تاكدت انه تم تحميل و تنصيب الملف بنجاح.
اذهب الى
...Tools > Add-in Manager

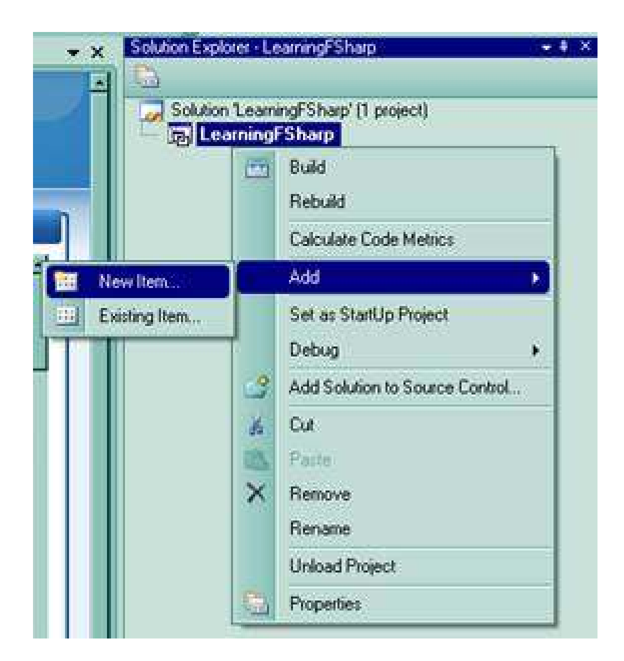


٧- فى حالة عدم وجود "F# Interactive for Visual Studio" كما فى الشكل التالى فهذا معناة وجود مشكلة فى الملف الذى قمت بتحميلة مسبقا فى الخطوة رقم ١ .

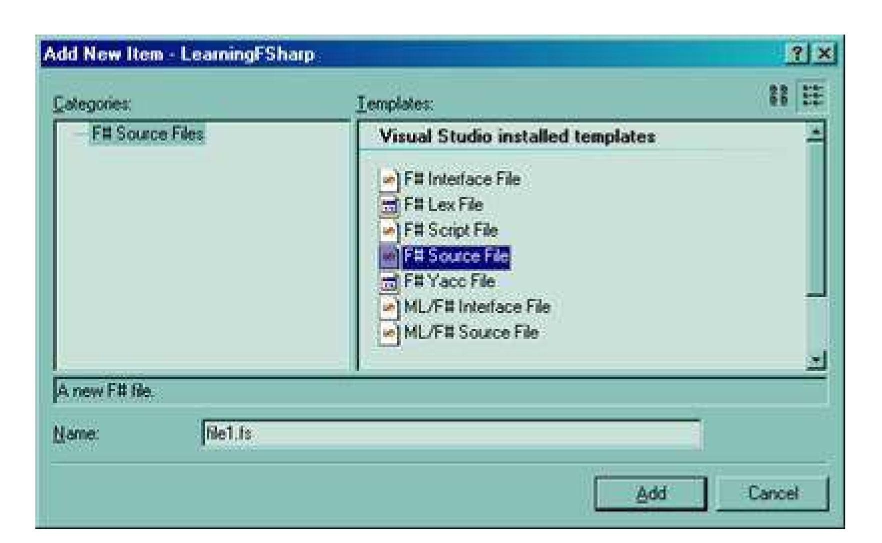
٨- قم باختيار "F# Interactive for Visual Studio" و اختار الخانتين ثم اضغط موافق .



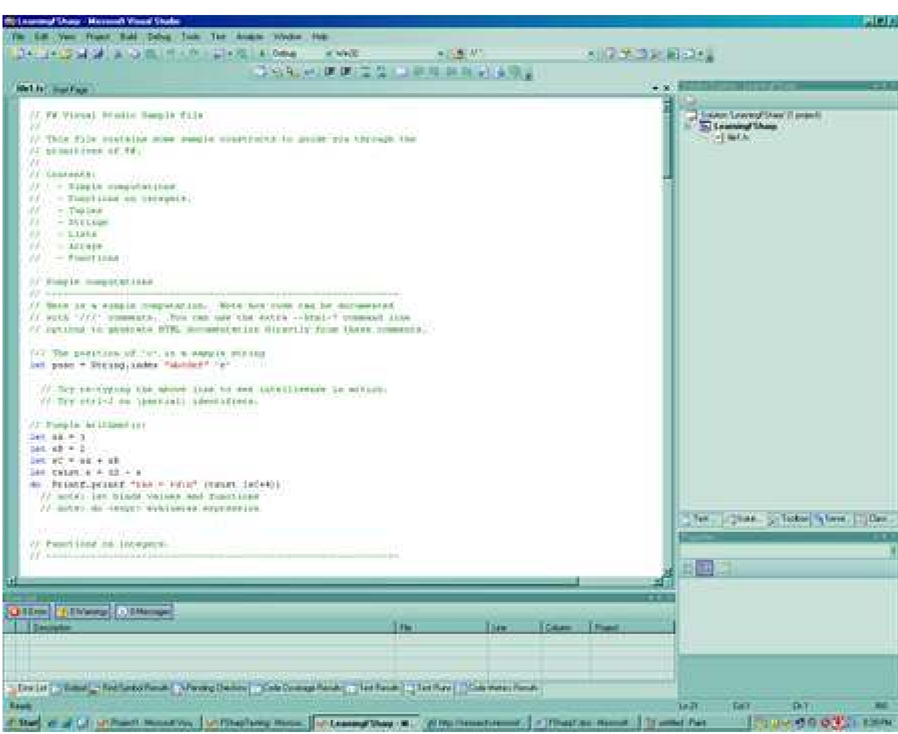
٩- فى الصفحة الرئيسية لمشروعك باستخدام لغة f# قم باضافة عنصر جديد لصفحتك كما فى الصورة التالية .

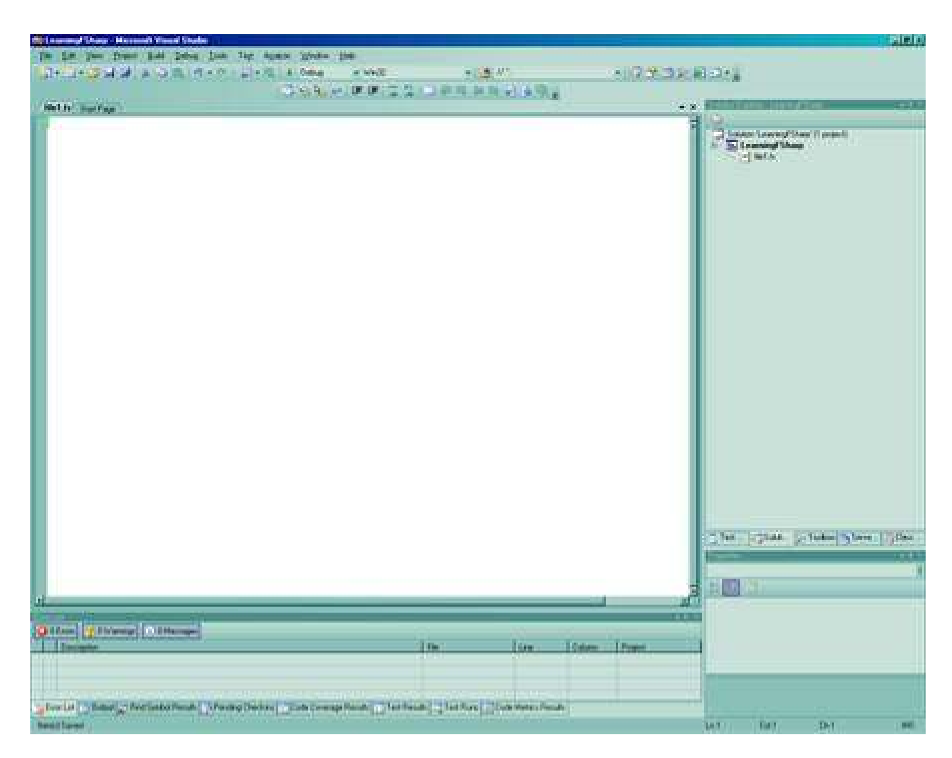


١٠- اختار f# source file



١٠- سيظر لك صفحة جديدة بامتداد .fs* التى ستقوم باستخدامها فى البرمجة مثل لغات ال(vb,c#,c++) كما فى الشكل التالى .





تعريف ببعض قواعد لغة F#

تعريفات:
IDENTIFIERS المسميات هي الأسماء التي تطلق على المتغيرات والدوال و التي تتكون من مجموعه من الحروف والأرقام حيث يمكن لاحد المسميات ان يصل طولها لغايه 32 حرف ولكن الأستخدام يفضل ان لا يتجاوز طول المسمى عن ثمانيه رموز حيث ان المترجم لا يأخذ في اعتباره سوى الثمانيه الرموز الأولى من اسم المتغير
<P>CHAR<P>حرف</P> <P>INT<P>رقم صحيح قصير</P> <P>LONG<P>رقم صحيح طويل</P> <P>FLOAT<P>رقم حقيقي اي به علامه عشريه</P> <P>DOUBLE<P>رقم حقيقي مضاعف به علامه عشريه</P> الفلوت عبارة عن رقم فلوتنج بوينت ولكن single precision يعني تخزن الرقم في مساحة 32 بت فقط أما ال double precision تخزن الرقم في مساحة 64 بت (بت للعلامة (سالب أو موجب)، وثمانية بت للرقم الصحيح، والباقي للكسور)

فى الجزء التالى سنقوم باستخدام واجهة ال F# فى التعامل مع الاوامر و ستلاحظ ان لغة F# ليست من اللغات الثقيلة التى تطلب منك تعريف كل متغير (رقم ، نص....) لكن تقوم بنفسها بتحديد نوع المتغير بعد ادخال قيمته .

١- للبدء ادخل على

Start | All Programs | Microsoft F# CTP 1.9.6.2 | F# Interactive (Console):

ستظهر لك النافذة التالية .

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

>
```

شكل رقم (١)

١- اكتب الامر التالى علما بان نهاية سطر الاوامر ال f# تنتهى ب " ;; "

F#

```
42;;
```

سيقوم محرر ال f# بتحديد طبيعة المعرف و سوف يضع لها IDENTIFIERS باسم "it".

بعد ادخال القيمة ٤٢ و هية قيمة رقمية تم التعرف عليها على انها (رقم صحيح int) بقيمة (42)

محفوظة فى المعرف (it) .

كما هو موضوع بالصورة رقم ٢.

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> 42;;
val it : int = 42
>
```

شكل رقم (2)

اكتب الامر التالى ثم اضغط Enter

سيتم حفظ القيمة فى "it" و يقوم هذا المعرف با خذ جميع القيم التالية فى نفس ال session الى ان يتم اغلاقها .

F#

```
it;;
```

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> 42;;
val it : int = 42
> it;;
val it : int = 42
>
```

شكل رقم (3)

اكتب الامر التالى ثم اضغط Enter

F#

```
42.0;;
```

باضافة خانه عشرية الى الرقم ٤٢ سيتم التعرف علية على انه <P>FLOAT رقم حقيقي اي به علامه عشريه</P>

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> 42;;
val it : int = 42
> it;;
val it : int = 42
> 42.0;;
val it : float = 42.0
>
```

اكتب الامر التالى ثم اضغط Enter

F#

```
"42";;
```

بعد وضع القيمة فى اقتباس مزدوج " " سيتم التعرف علية على انه نص(string) سواء كان مجموعةحروف او ارقام.

C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe

```
Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> 42;;
val it : int = 42
> it;;
val it : int = 42
> 42.0;;
val it : float = 42.0
> "42";;
val it : string = "42"
>
```

شكل رقم (٥)

١- التعامل باستخدام

Tuples

يُمكِن ال (**tuples**) المبرمجين من الاستعانه بعدة انواع من البيانات فى حقل واحد مثل (رقم، تاريخ ،حروف.....).
اكتب الامر التالى ثم اضغط Enter

F#
```
(42, "Hello F#");;
```

تم ادخال قيمتين قيمة رقم صحيح و قيمة نصيه على هيئة ("int * string")

و تم التعرف عليها على ان القيمة الاولى (**integer**) و الثانية (**string**) .

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe

Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> 42;;
val it : int = 42
> it;;
val it : int = 42
> 42.0;;
val it : float = 42.0
> "42";;
val it : string = "42"
> (42, "Hello F#");;
val it : int * string = (42, "Hello F#")
>
```

شكل رقم (٦)

يُمكن لل (**tuples**) ان يحتوى على عدة قيم و ليس فقط زوجين من القيم .

اكتب الامر التالى ثم اضغط Enter

F#
```
(42, "Hello F#", 42.0);;
```

فى المثال السابق تم ادخال ٣ قيم مختلفة و هم بالترتب كالتالى :

"int * string * float"

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe

Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> 42;;
val it : int = 42
> it;;
val it : int = 42
> 42.0;;
val it : float = 42.0
> "42";;
val it : string = "42"
> (42, "Hello F#");;
val it : int * string = (42, "Hello F#")
> (42, "Hello F#", 42.0);;
val it : int * string * float = (42, "Hello F#", 42.0)
>
```

شكل رقم (٧)

٢- استخدام الدوال (Functions) كقيم

فى لغة ال #f لا يوجد اى فرق ما بين الدوال (functions) و القيم (values) لان الدوال عبارة عن قيم . و يمكن اعطاء القيمة لمعرف مثل اى قيمة اخرى.

اكتب الامر التالى ثم اضغط Enter

F#

```
fun x -> x + 1;;
```

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> 42;;
val it : int = 42
> it;;
val it : int = 42
> 42.0;;
val it : float = 42.0
> "42";;
val it : string = "42"
> (42, "Hello F#");;
val it : int * string = (42, "Hello F#")
> (42, "Hello F#", 42.0);;
val it : int * string * float = (42, "Hello F#", 42.0)
> fun x -> x + 1;;
val it : int -> int = <fun:clo@0>
>
```

شكل رقم (٨)

و بما ان اخر قيمة مسندة الى "it" كانت مضاف اليها ١+

x + 1;;

نقوم باسناد قيمة ٤١ الى "it" و التى ترمز الى x (الدالة السابقة)

اكتب الامر التالى ثم اضغط Enter

F#

```
it 41;;
```

ستكون النتيجة النهائية ٤١+١ = ٤٢

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> 42;;
val it : int = 42
> it;;
val it : int = 42
> 42.0;;
val it : float = 42.0
> "42";;
val it : string = "42"
> (42, "Hello F#");;
val it : int * string = (42, "Hello F#")
> (42, "Hello F#", 42.0);;
val it : int * string * float = (42, "Hello F#", 42.0)
> fun x -> x + 1;;
val it : int -> int = <fun:clo@0>
> it 41;;
val it : int = 42
>
```

شكل رقم (٩)

سنقوم باختبار اخر قيمة ل "it "

F#

```
it;;
```

نحقق لنا هنا ان "it" لم يعد يحتوى قيمة الدالة الاول 'x'و التى كانت قيمتها ٤١ و اصبحت قيمتها الان ٤٢

و بذ الك سوف نقوم فى جزء اخر من الشرح بالتعرف على طريقة اخرى لحفظ قيمة الدوال دون تغيير.

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> 42;;
val it : int = 42
> it;;
val it : int = 42
> 42.0;;
val it : float = 42.0
> "42";;
val it : string = "42"
> (42, "Hello F#");;
val it : int * string = (42, "Hello F#")
> (42, "Hello F#", 42.0);;
val it : int * string * float = (42, "Hello F#", 42.0)
> fun x -> x + 1;;
val it : int -> int = <fun:clo@0>
> it 41;;
val it : int = 42
> it;;
val it : int = 42
>
```

شكل رقم (١٠)

٣- استخدام الكلمة المحجوزة "Let" فى لغة F#

لا تحتوى لغة F# على متغيرات يمكن اسناد القيم اليها لذالك سنستخدم المعرف الاساسى "let" الذى يسمح باسناد قيمة الى مُعرف.

اكتب الامر التالى ثم اضغط Enter

F#

```
let a = 42;;
```

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> let a = 42;;

val a : int

>
```

شكل رقم (١١)

ففى المثال السابق تم اسناد قيمة رقمة الى المتغير a .

و يمكننا التاكد من قيمة a بكتابة الامر التالى .

F#

```
a;;
```

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> let a = 42;;

val a : int

> a;;
val it : int = 42
>
```

شكل رقم (١٢)

فى لغة ال #f بعد اعطاء قيمة معينه لمتغير فانها تظل ثابته لا تتغير .

(i.e. “let x = 42;;”)

و فى حالة اعطاءة قيمة جديدة يحتفظ بالقديمة لكنه يقوم باستخدام القيمة الجديدة فقط .

(i.e. “let x = ‘forty two’”)

يمكن استخدام المُعرف فى العمليات الحسابية مثل ;;a+1

حيث ان اخر قيمة ل a كانت ٤٢ و سوف نقوم بعملية جمع بالاضافة ١ الى قيمتها الاساسية .

F#

```
a + 1;;
```

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe

Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> let a = 42;;

val a : int

> a;;
val it : int = 42
> a + 1;;
val it : int = 43
>
```

شكل رقم (١٣)

لا حظ انك بالرغم انه تم اسناد قيمة جديدة للمتغير a فانه يقوم باستخدامها و لكنه لا زال يحتفظ بقيمته السابقة و لكن لا يمكن اعادة استخدامها مرة اخرى او الوصول اليها.

F#

```
let a = 0;;
a;;
```

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe

Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> let a = 42;;

val a : int

> a;;
val it : int = 42
> a + 1;;
val it : int = 43
> let a = 0;;

val a : int

> a;;
val it : int = 0
>
```

شكل رقم (١٤)

ايضا يمكن اسناد قيمة جديدة لا تتعلق بنوعية القيمة السابقة

حيث فى المثال السابق قيمة a كانت رقمة و تقدر ب ٤٢ و لكننا فى المثال التالى سنقوم باسناد قيمة نصية اليها string

F#

```
let a = “42”;;
```

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> let a = 42;;

val a : int

> a;;
val it : int = 42
> a + 1;;
val it : int = 43
> let a = 0;;

val a : int

> a;;
val it : int = 0
> let a = "42";;

val a : string

>
```

شكل رقم (١٥)

٤- دمج ال (tuples) مع (identifiers)

يمكن اسناد عدة قيم الى معرف واحد

مثل ان يكون اول حقل للمعرف رقم و ثانى حقل نص و هكذا

فى المثال التالى سنقوم باعطاء المعرف a قيمة رقمية ٤٢ و قيمة نصية **“Hello F#”**

على هيئة **int * string** بالترتيب .

F#

```
let a = (42, “Hello F#”);;
```

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
For help type #help;;
> let a = 42;;
val a : int
> a;;
val it : int = 42
> a + 1;;
val it : int = 43
> let a = 0;;
val a : int
> a;;
val it : int = 0
> let a = "42";;
val a : string
> let a = (42, "Hello F#");;
val a : int * string
>
```

شكل رقم (١٦)

يمكن عمل فواصل فى حالة وجود قيمتان عن طريق الدالتين ***fst and snd*** بحيث نقوم باستخدام اول قيمة فى معرف جديد و ثانى قيمة فى معرف اخر جديد .

كما فى المثال التالى :

F#

```
let b = fst a;;
let c = snd a;;
```

حيث قمنا باعطاء المعرف الاول قيمة رقيمة و المعرف التانى قيمة نصية .

و يمكننا مشاهدة القيم التى تم اسنادها اليهم عن طريق الامر التالى:

F#

```
b;;
c;;
```

C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe

```
val a : int

> a;;
val it : int = 0
> let a = "42";;

val a : string

> let a = (42, "Hello F#");;

val a : int * string

> let b = fst a;;

val b : int

> let c = snd a;;

val c : string

> b;;
val it : int = 42
> c;;
val it : string = "Hello F#"
>
```

شكل رقم (١٧)

يمكن ايضا اعادة اسناد او توزيع القيم الى المعرف الاساسى a

مثال :

F#

```
let (b, c) = a;;
b;;
c;;
```

C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe

```
val a : int * string

> let b = fst a;;

val b : int

> let c = snd a;;

val c : string

> b;;
val it : int = 42
> c;;
val it : string = "Hello F#"
> let (b, c) = a;;

val c : string
val b : int

> b;;
val it : int = 42
> c;;
val it : string = "Hello F#"
>
```

شكل رقم (١٨)

فى حالة استخدام اكثر من قيمة فى ال tuple لن نستطيع استخدام الدالتين **fst** and **snd**

و عند محاولة استخدامهم بعد زيادة عدد القيم عن قيمتين سيحدث خطئ و تظهر رسائل خطئ

سنقوم هنا باضافة ٣ قيم الى ال tuple a عبارة عن **integer**, a **string** and a **float**. على التوالى

F#

```
let a = (42, “Hello F#”, 42.0);;
```

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
val b : int
> let c = snd a;;
val c : string
> b;;
val it : int = 42
> c;;
val it : string = "Hello F#"
> let (b, c) = a;;
val c : string
val b : int
> b;;
val it : int = 42
> c;;
val it : string = "Hello F#"
> let a = (42, "Hello F#", 42.0);;
val a : int * string * float
>
```

شكل رقم (١٩)

ثم عند محالة استخدام الدالتين ***fst** and **snd*** سيظهر الخطئ التالى :

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
> let a = (42, "Hello F#", 42.0);;
val a : int * string * float
> let b = fst a;;
  let b = fst a;;
  ----------------^^
stdin(16,13): error FS0001: Type mismatch. Expecting a
        int * string
but given a
        int * string * float.
The tuples have differing lengths of 2 and 3.
> let c = snd a;;
  let c = snd a;;
  ----------------^^
stdin(17,13): error FS0001: Type mismatch. Expecting a
        int * string
but given a
        int * string * float.
The tuples have differing lengths of 2 and 3.
>
```

شكل رقم (٢٠)

سنقوم بالتالى باسناد كل جزء على حدى كما فى المثال التالى :

F#

```
let (b, c, d) = a;;
```

فنكون حصلنا على قيم مسندة لكل جزء فى ال tuple

كما فى الشكل التالى .

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
  let b = fst a;;
  -------------^^

stdin(16,13): error FS0001: Type mismatch. Expecting a
        int * string
but given a
        int * string * float.
The tuples have differing lengths of 2 and 3.
> let c = snd a;;

  let c = snd a;;
  -------------^^

stdin(17,13): error FS0001: Type mismatch. Expecting a
        int * string
but given a
        int * string * float.
The tuples have differing lengths of 2 and 3.
> let (b, c, d) = a;;

val d : float
val c : string
val b : int

>
```

شكل رقم (٢١)

يمكن التاكد من قيمة كل عنصر كما فى المثال التالى :

F#

```
b;;
c;;
d;;
```

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
        int * string * float.
The tuples have differing lengths of 2 and 3.
> let c = snd a;;

  let c = snd a;;
  -------------^^

stdin(17,13): error FS0001: Type mismatch. Expecting a
        int * string
but given a
        int * string * float.
The tuples have differing lengths of 2 and 3.
> let (b, c, d) = a;;

val d : float
val c : string
val b : int

> b;;
val it : int = 42
> c;;
val it : string = "Hello F#"
> d;;
val it : float = 42.0
>
```

شكل رقم (٢٢)

يمكن ايضا حذف قيم من ال tuple

سنقوم فى هذا المثال بالابقاء على قيمة واحدة و حذف الباقى فنقوم بوضع _ مكان القيم التى لا نريدها

ثم نقوم بالاسناد الى القيم الاخرى قمنا باسناد ثانى قيمة الى (e)

F#

```
let (_, e, _) = a;;
```

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
  let c = snd a;;
  ---------------^^

stdin(17,13): error FS0001: Type mismatch. Expecting a
        int * string
but given a
        int * string * float.
The tuples have differing lengths of 2 and 3.
> let (b, c, d) = a;;

val d : float
val c : string
val b : int

> b;;
val it : int = 42
> c;;
val it : string = "Hello F#"
> d;;
val it : float = 42.0
> let (_, e, _) = a;;

val e : string

>
```

شكل رقم (٢٣)

و عند مشاهدة القيمة المسندة الى e وجدناها القيمة الثانية لل tuple

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe
stdin(17,13): error FS0001: Type mismatch. Expecting a
        int * string
but given a
        int * string * float.
The tuples have differing lengths of 2 and 3.
> let (b, c, d) = a;;

val d : float
val c : string
val b : int

> b;;
val it : int = 42
> c;;
val it : string = "Hello F#"
> d;;
val it : float = 42.0
> let (_, e, _) = a;;

val e : string

> e;;
val it : string = "Hello F#"
>
```

شكل رقم (٢٤)

الدوال

ولان لغة ال #f تتعامل مع الدوال على انها ببساطة مجموعة من البيانات المختلفة فهذ ا يسهل علينا جدا التعامل مع الدوال.

و سنقوم فى الخطوات التالية بشرح كيف تتعامل لغة ال #f مع الدوال .

اسناد الدالة الى مُعرف

فى هذا المثال سنتعلم كيف ان الدوال عبارة عن قيم يمكن اسنادها الى مُعرف .

سنقوم ببناء دالة و سنقوم باسناد قيمتها الى مُعرف (**addTenToNumber**) .

```
F#
let addTenToNumber = (fun x -> x + 10);;
```

فى المثال السابق قمنا باسناد المعرف (**addTenToNumber**) الى الدالة حيث ان تعريف الدالة الاساسى (**fun**)

ثم اسم المتغير(x)

ثم السهم الذى يدل على القيمة (**->**) ثم فى النهاية محتوى الدالة (**x + 10**).

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe

Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> let addTenToNumber = (fun x -> x + 10);;

val addTenToNumber : int -> int

>
```

شكل رقم (٢٥)

بعد تشغل الامر السابق بلغة *#f* ظهر لنا ان المعرف ***addTenToNumber*** تم تقديرة انه رقم صحيح

و ياخذ قيم لارقام صحيحة و يعطى نتيجته برقم صحيح (***int -> int***)

سنقوم الان باعطاء قيمة للمعرف(x) الموجود فى الدالة (**addTenToNumber**) التى قمنا ببنائها فى الخطوة السابقة

سنقوم باعطائة قيمة ٣٢ كما فى المثال التالى .

F#

```
addTenToNumber 32;;
```

```
C:\Program Files\FSharp-1.9.6.2\bin\fsi.exe

Microsoft F# Interactive, (c) Microsoft Corporation, All Rights Reserved
F# Version 1.9.6.2, compiling for .NET Framework Version v2.0.50727

Please send bug reports to fsbugs@microsoft.com
For help type #help;;

> let addTenToNumber = (fun x -> x + 10);;

val addTenToNumber : int -> int

> addTenToNumber 32;;
val it : int = 42
>
```

ختام

احتسب عند الله ان يكون هذا الكتاب اول كتاب عربى فى لغة F#
ترجم فقراته من مصادر مختلف و لقلة المصادر اكتفيت ان يكون مجرد مقدمة
لمن يريد ان يتعرف على طبيعة هذة اللغة التى سيكون لها شان عظيم فى المستقبل.

و انتظروا باذن الله الكتاب التالى عند اكتمال المصادر

(F# من البداية حتى الاحتراف)

ولا تنسونى من صالح دعائكم